AF
IFW

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Appellants:	Novaes et al.	Group Art Unit: 2155
Serial No.:	09/584,259	Examiner: Michael Young Won
Filed:	05/31/00	Appeal No.:
Title:	METHOD, SYSTEM AND PROGRAM PRODUCTS FOR MANAGING PROCESSING GROUPS OF A DISTRIBUTED COMPUTING ENVIRONMENT	

Certificate of Mailing

I hereby certify that this correspondence is being deposited with the U.S. Postal Service as first class mail in an envelope addressed to: Mail Stop Appeal Briefs – Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on November 1, 2004.

Jill K. Becker

Jill K. Becker

Date of Signature: November 1, 2004.

Mail Stop Appeal Briefs – Patents
Commissioner for Patents
P.O. Box 1450, Alexandria, VA 22313-1450

11/04/2004 WASFAW1 00000020 090463 09584259
02 FC:1402 340.00 DA

Brief of Appellants

Dear Sir:

This is an appeal from a final rejection, mailed March 16, 2004, rejecting claims 1-49, all the claims being considered in the above-identified application. The Appeal Brief was due within two months from the date the Notice of Appeal was received at the United States Patent and Trademark Office. Since appellants' postcard indicates that the Notice of Appeal was received on August 6, 2004, this Brief was due on or before October 6, 2004, and thus, a Request For Extension Of Time and the requisite fee are enclosed herewith. This Brief is accompanied

by a transmittal letter authorizing the charging of appellants' deposit account for payment of the requisite fee set forth in 37 C.F.R. §1.17(c).

Appellants' brief is being filed after the effective date of the final BPAI Rules, September 13, 2004, and, therefore, the format and content of appellants' brief is in compliance with the requirements set forth in 37 CFR §41.37(c). If appellants' brief does not comply with the requirements set forth in 37 CFR §41.37(c), appellants request notification of the reasons for noncompliance and the opportunity to file an amended brief pursuant to 37 CFR §41.37(d).

Real Party in Interest

This application is assigned to International Business Machines Corporation by virtue of an assignment executed by the co-inventors on September 6, 2000, September 7, 2000, September 22, 2000, and September 25, 2000; and recorded with the United States Patent and Trademark Office at reel 011185, frame 0283, on September 29, 2000. Therefore, the real party in interest is International Business Machines Corporation.

Related Appeals and Interferences

To the knowledge of the appellants, appellants' undersigned legal representative, and the assignee, there are no other appeals or interferences, which will directly affect or be directly affected by or have a bearing on the Board's decision in the instant appeal.

Status of Claims

This patent application was filed on May 31, 2000 with the United States Patent and Trademark Office. As filed, the application included six (6) claims, three (3) of which were independent claims (i.e., claims 1, 3, 5).

A preliminary amendment was filed on April 12, 2002 adding claims 7-48, six (6) of which were independent claims (i.e., claims 22, 24, 26, 28, 35, 42).

In an initial Office Action, dated September 5, 2003, claims 1-48 were rejected under 35 U.S.C. 102(e) as being anticipated by Moiin (U.S. Patent No. 6,108,699; hereinafter, Moiin). In appellants' response mailed January 5, 2004, claims 1, 3, 5, 7, 12, 17, 22, 24, 26, 28, 29, 35, 36, 42 and 43 were amended and dependent claim 49 was added.

In a second and final Office Action, dated March 16, 2004, claims 1, 3, 5, 28, 35 and 42 were rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement; claims 1-21 and 28-49 were rejected under 35 U.S.C. 102(e) as being anticipated by Moiin; and claims 22-27 were rejected under 35 U.S.C. 102(e) as being anticipated by Shrivastava et al. (U.S. Patent No. 6,449,734; hereinafter Shrivastava). In appellants' response mailed May 17, 2004, claims 1, 3, 5, 28, 35, and 42 were amended.

Appellants received an Advisory Action, dated July 21, 2004, which indicated that appellants' amendments had been entered, but the response mailed May 17, 2004 did not place the application in condition for allowance.

A Notice of Appeal to the Board of Patent Appeals and Interferences was mailed on August 3, 2004, with the requisite fee for a one-month extension of time. The Notice of Appeal was received at the United States Patent and Trademark Office on August 6, 2004. The status of the claims is therefore as follows:

Claims allowed – none;

Claims objected to – none;

Claims rejected – 1-49; and

Claims canceled – none.

Appellants are appealing the rejection of claims 1-49.

Status of Amendments

The claim amendments included in appellants' Response to Final Office Action dated May 17, 2004 have been entered by the Examiner. The claims as set out in Appendix A include all entered amendments.

Summary of Claimed Subject Matter

In one aspect of the invention, appellants' claim a method (independent claim 1), system (independent claim 3) and program storage device (independent claim 5) for managing processing groups of a distributed computing environment. The method includes, for instance, comparing at least a portion of an individual prospective member state of a prospective member of a processing group with at least a portion of a group state of the processing group, the individual prospective member state comprising state defined for the individual prospective member (see, e.g., 906, FIG. 9A; page 18, lines 1-10); updating the at least a portion of the individual prospective member state, should the comparing indicate a difference (see, e.g., 910, FIG. 9A; page 18, lines 11-18); and joining the prospective member to the processing group, in response to the updating (see, e.g., 912, FIGs. 9A; 922-926, FIG. 9B; page 19, lines 16-21; page 20, lines 12-23).

As a further aspect, appellants recite in dependent claims 7, 12 and 17 that the individual prospective member state comprises a sequence number of the prospective member, and wherein the comparing comprises comparing the sequence number of the prospective member with the sequence number of the processing group (see, e.g., 906, FIG. 9A; page 18, lines 1-11).

As yet a further aspect, dependent claims 8, 13 and 18 recite that the sequence number of the prospective member is less than the sequence number of the processing group, and wherein

the updating includes updating the sequence number of the prospective member with the sequence number of the processing group (e.g., 906, 910, FIG. 9A; page 18, lines 5-15).

In yet another embodiment, dependent claims 9, 14 and 19 recite determining an activity status of the processing group prior to the updating of the sequence number, wherein the updating the sequence number of the prospective member includes updating if the processing group is active (e.g., 908, FIG. 9A; page 18, lines 7-15).

As a further aspect, dependent claims 11, 16 and 21 recite wherein the updating at least a portion of the state of the processing group after the joining includes updating the sequence number of the processing group (see, e.g., 922, FIG. 9B; page 20, lines 12-17).

In yet a further embodiment, dependent claim 49 recites wherein the sequence number of the prospective member identifies a version of a proposed processing group to join (see, e.g., pages 14-15, 18-20).

Independent claim 3 and dependent claims 12-14 and 16 include means plus function language. The functions are described above, including reference to the specification and drawings. This logic is performed by nodes, such as, for example, those depicted in FIG. 8.

In another aspect of the invention, appellants' claim a method (independent claim 22), system (independent claim 24) and program storage device (independent claim 26) for managing processing groups of a distributed computing environment. The method includes, for instance, detecting a failure of at least one member of a processing group (see, e.g., 1500, FIG. 15; page 20, lines 24-27); quiescing activity to a group state of the processing group (see, e.g., 1507, FIG. 15; page 21, lines 4-6); and updating at least a portion of the group state in order to exclude the at least one member of the processing group, wherein the updating comprises updating a sequence number of the group state, said sequence number identifying a version of the processing group (see, e.g., 1508, FIG. 15; page 21, lines 6-16).

Independent claim 24 includes means plus function language. The functions are described above, including reference to the specification and drawings. The logic is performed by nodes, such as, for example, those depicted in FIG. 8.

In another aspect of the invention, appellants' claim a method (independent claim 28), system (independent claim 35) and program storage device (independent claim 42) for managing processing groups of a distributed computing environment. The method includes, for instance, joining a prospective member to an inactive processing group (see, e.g., 616, FIG. 6B; page 14, lines 14-19); comparing at least a portion of an individual prospective member state with at least a portion of a group state of the processing group, said individual prospective member state comprising state defined for the individual prospective member (see, e.g., 618, FIG. 6B; page 14, lines 20-24); and updating the at least a portion of the group state (see, e.g., 620, FIG. 6B; page 14, lines 20-24).

As a further embodiment, in dependent claims 29, 36 and 43, appellants claim wherein the individual prospective member state includes a sequence number of the prospective member, and wherein the comparing comprises comparing the sequence number of the prospective member with the sequence number of the group state (see, e.g., 618, FIG. 6B; page 14, lines 14-19).

In yet a further aspect, in dependent claims 30, 37 & 44, the updating includes updating a sequence number of the group state with the sequence number of the prospective member if the sequence number of the prospective member is smaller than the sequence number of the group state (see, e.g., 618-620, FIG. 6B; page 14, lines 20-24).

In yet a further aspect, in dependent claims 31, 38 and 45, appellants recite that the updating includes updating the sequence number of the group state with the highest sequence number of the members of the processing group if a quorum of the processing group exists (see, e.g., 622-624, FIG. 6B; page 15, lines 1-7).

As yet a further embodiment, in dependent claims 33, 40 and 47, appellants recite that the activating includes updating a local copy of the group state for any member of the processing group whose sequence number is less than a current sequence number of the processing group (see, e.g., 630-632, FIG. 6C; page 15, lines 15-22).

In yet a further embodiment, in dependent claims 34, 41 and 48, appellants recite that the activating further comprises changing the group state to active if a majority of the members of the processing group have a sequence number matching the current sequence number and none of the members has aborted (see, e.g., 644, FIG. 6D; page 16, lines 11-16).

Independent claim 35 and dependent claims 36-41 include means plus function language. The functions are described above, including reference to the specification and drawings. This logic is performed by nodes, such as those depicted in FIG. 6A.

Grounds of Rejection to Be Reviewed On Appeal

1. Claims 1-21 and 28-49 stand rejected under 35 U.S.C. 102(e) as being anticipated by Moiin, U.S. Patent No. 6,108,699; and

2. Claims 22-27 stand rejected under 35 U.S.C. 102(e) as being anticipated by Shrivastava et al., U.S. Patent No. 6,449,734.

Argument

I. Rejection under 35 U.S.C 102(e) Over US Patent No. 6,108,699 to Moiin

A. Claims 1-6, 8-21 and 28-48:

Claims 1-6, 8-21 and 28-48 stand rejected under 35 U.S.C. 102(e) as being anticipated by Moiin, U.S. Patent No. 6,108,699. Appellants respectfully submit that the rejection of these claims is erroneous and respectfully request reversal of this rejection for the reasons below.

1. Independent Claims 1, 3 and 5:

Appellants' invention is directed, in one aspect, to a protocol used to join a prospective member to a processing group. The join protocol includes various steps taken to ensure configuration consistency. These steps include, for instance, the comparing and updating of state to appropriately join the prospective member to the group. One example of the state used in appellants' protocol is individual prospective member state (i.e., state specific to the prospective member). In one instance, this state is a sequence number of the prospective member that is used throughout the join protocol to control whether a prospective member joins the group and to ensure configuration consistency.

As one particular example, appellants claim a method of managing processing groups of a distributed computing environment (e.g., independent claim 1). The method includes, for instance, comparing at least a portion of an individual prospective member state of a prospective member of a processing group with at least a portion of a group state of the processing group, the individual prospective member state comprising state defined for the individual prospective member; updating the at least a portion of the individual prospective member state, should the comparing indicate a difference; and joining the prospective member to the processing group, in response to the updating. Thus, in this aspect of appellants' claimed invention, particular steps are performed in order to join a prospective member to the processing group. These steps include comparing individual prospective member state to group state, updating the individual prospective member state, if there is a difference in the comparing, and joining the prospective member to the group in response to the update. These particular steps claimed by appellants are very different from the teachings of Moiin.

Although Moiin has a join protocol, the join protocol of Moiin is quite distinct from the join protocol of appellants. Unlike appellants' join protocol, Moiin's join protocol includes sending reconfiguration messages to nodes, waiting for return messages from those nodes and then making comparisons of the messages. If a comparison shows a difference, then the join

protocol is aborted. This is described in Cols. 5-7 of Moiin. This is very different from appellants' claimed invention in which individual prospective member state is compared to group state. If the comparison indicates a difference, the individual prospective member state is updated and the member joins the group, in response to the updating. Thus, unlike in Moiin, in appellants' claimed invention, if the comparison indicates a difference, the state is updated and the join continues. This is in contrast to Moiin, in which the join is aborted, if a comparison of the messages indicates a difference.

Moiin fails to describe, teach or suggest the specific elements claimed by appellants. In particular, the join protocol of Moiin fails to describe, teach or suggest one or more of appellants' claimed elements. For example, Moiin fails to describe, teach or suggest updating at least a portion of the individual prospective member state should a comparison indicate a difference and joining the prospective member to the group, in response to the updating. Instead, in Moiin, messages are broadcast, return messages are received and compared, and if any differences are found in the comparisons, which are also different than appellants' comparisons, then the join protocol fails. This is opposite to that claimed by appellants.

There are various message comparison steps in Moiin's join protocol, but the result of each one is an abort of the join protocol, and not a continuation of the protocol, as claimed by appellants. For example, at Test 418 (FIG. 4, Col. 7, lines 31-40), a determination is made as to whether reconfiguration messages have been received from all the petitioned nodes. If not, processing is aborted (FIG. 4, #420). That is, if the message comparison shows a difference, then the join does not take place.

Similarly, at Test 422 (FIG. 4, Col. 7, lines 44-52), a comparison is made as to whether all the received reconfiguration messages are identical. If not, then processing is aborted (FIG. 4, #420). Again, if the test indicates a difference, the join does not take place. This is opposite to that claimed by appellants in which if the comparison indicates a difference, an update to state is made and the join continues. This difference between Moiin and appellants' claimed

invention is attributable to the fact that the protocols are very distinct from one another. Moiin uses reconfiguration messages to determine whether a join takes place, while appellants use a comparison between states (e.g., sequence numbers) to make the determination.

Another test in the join protocol of Moiin is a test of whether the number of nodes responding to the initial broadcast message is greater than 1 (see, test step 410, FIG. 4, Col. 6, lines 33-42). However, this comparison is not a comparison between group state and individual state, as claimed by appellants. It is simply a comparison to the number one. Further, state is not updated in response to the comparison. Instead, messages are re-broadcast and processing continues from there.

A further test in the join protocol is whether the joining node is an isolated node, STEP 412 (FIG. 4, Col. 6, lines 43-56). If the node is not isolated, as determined by human intervention, then messages are broadcast to the petitioned nodes. If the node is isolated, then the join is aborted. This isolation determination step is very different from appellants' claimed invention of comparing at least a portion of individual prospective member state with at least a portion of group state, updating the at least a portion of the individual prospective member state should the comparing indicate a difference, and joining the prospective member to the processing group in response to updating. Moiin fails to describe these steps in joining a node to a cluster. Instead, Moiin uses a very different protocol in which reconfiguration messages are broadcast and the responses to those messages are compared.

There is no description, teaching or suggestion in Moiin of appellants' step-wise protocol to join a prospective member to a processing group. That is, there is no description, teaching or suggestion of comparing at least a portion of an individual prospective member state of a prospective member of a processing group with at least a portion of group state of the processing group; updating the at least a portion of the individual prospective member state, should the comparing indicate a difference, and joining the prospective member to the processing group, in response to the updating. In contrast, as one example, the failure of a message comparison in

Moiin's protocol dictates aborting the join protocol, rather than updating state and continuing with the join, as claimed by appellants.

In addition to the above, appellants respectfully submit that the join protocol of Moiin also fails to employ an individual prospective member state, as claimed by appellants. Instead, Moiin only uses group state. That is, the comparisons and updating in Moiin are of group state. There is no description, teaching or suggestion of comparing and/or updating individual member state, as claimed by appellants.

With the join protocol of Moiin, a node joins a cluster by forming a new prospective cluster. Once all of the prospective cluster members are in agreement, the cluster is formed. In particular, as described with reference to FIG. 4 and Cols. 5-7 of Moiin, the join process of Moiin includes broadcasting reconfiguration messages to nodes 1-5; waiting to receive replies to those reconfiguration messages; and eventually updating various group states, including a next cluster size field, a next cluster vector, a cluster size field and a cluster vector.

Moiin's process is quite distinct from appellants' claimed invention. In appellants' claimed invention, to join a cluster, a comparison is made between individual prospective member state and group state, and then based on that comparison, the individual prospective member state is updated. For instance, in appellants' claimed invention, individual prospective member state (e.g., an individual sequence number), such as state that once located at the individual prospective member represents only that member and not the group, is compared to group state that is stored, for instance, in persistent storage. This group state includes, for instance, a group sequence number. If the individual state (e.g., individual sequence number) is different than the group state (e.g., group sequence number), then the individual state (e.g., individual sequence number) is updated. No such steps are described in Moiin.

For example, while Moiin describes some comparisons, the comparisons are not between individual prospective member state and group state. As one instance, a comparison is

performed at STEP 410 of FIG. 4 in which a comparison of the number of proposed members for the clusters is compared to one. There is no comparison of an individual prospective member state with group state. Instead, the group state is merely compared to a particular value. As a further example, at INQUIRY 422, a determination is made as to whether all received reconfigure messages are identical. Thus, this comparison is to the messages themselves. There is no teaching or suggestion of comparing individual prospective member state with group state, as claimed by appellants.

In addition to the above, there is no teaching or suggestion of appellant's claimed element of updating said at least a portion of the individual prospective member state, should the comparison indicate a difference. Instead, in Moiin, if there is a difference between the messages received, then processing is aborted. There is no teaching or suggestion of updating the individual prospective member state should the comparing indicate a difference. Each of the states that is compared and updated defines the group. For instance, the cluster vector field describes all of the nodes of the cluster, and the cluster size field describes the size of the cluster. None of those fields includes state defined for the individual member, as claimed by appellants. The only individual state described in Moiin is the identification field, and this field is not compared or updated.

The comparing and updating of group state is described throughout Moiin. For instance, it is explicitly stated in Col. 6, lines 13-17:

> In step 408, CMM 220A (FIG. 3) updates next cluster size field 308 and next cluster vector 310 to represent a cluster which includes node 0 and all nodes from which CMM 220A receives a reconfiguration message in step 406 (FIG. 4).

It further states in Col. 6, lines 35-42:

> Specifically, in test step 410 (FIG. 4), CMM 220A (FIG. 3) compares the cluster size represented in cluster size field 304 to a value of one to determine whether any node other than node 0 is a

> member of the prospective cluster. If the cluster size is greater than one, processing transfers to step 414 (FIG. 4) which is described below. Conversely, if the cluster size is not greater than one, processing transfers to test step 412.

Yet further, Col. 7, lines 44-60 state:

> In test step 422, CMM 220A (FIG. 3) compares the received reconfiguration messages to determine whether all the received reconfiguration messages represent exactly the same cluster, i.e., whether all received reconfiguration messages agree as to cluster membership in the prospective cluster. If any of the received reconfiguration messages do not agree as to cluster membership, processing transfers from test step 422 (FIG. 4) to step 420 in which the reconfiguration of the cluster fails in the manner described above. Conversely, if all received reconfiguration messages agree as to membership in the prospective, processing transfers from test step 422 to step 424. In step 424, the prospective cluster is accepted and node 0 saves the prospective cluster as the current cluster by copying data from next cluster size field 308 (FIG. 3) and next cluster vector field 310 to cluster size field 304 and cluster vector field 306, respectively.

Thus, it is repeatedly stated in Moiin that the comparisons and updates are of group state, i.e., state that defines the group, rather than of individualized state of the prospective member. Again, the only individualized state described in Moiin is identification field 302 (FIG. 3), which is described in Col. 5, lines 33-35. Moiin states: "This identification field includes data which uniquely identifies node 0 and distinguishes node 0 from nodes 1-5 (FIG. 1)." Thus, while Moiin describes an identifier field, there is no description, teaching or suggestion of a comparison of the identifier field with the group state. Further, there is certainly no teaching or suggestion of updating the identifier field. Therefore, there is no description, teaching or suggestion in Moiin of comparing at least a portion of an individual prospective member state of a prospective member with the at least a portion of the group state, in which the individual prospective member state comprises state defined for the individual prospective member; nor is

there any description, teaching or suggestion of updating the individual prospective member state.

For all of the above reasons, appellants respectfully submit that Moiin does not anticipate independent claim 1, as well as independent claims 3 and 5. Therefore, appellants respectfully request reversal of the rejection of those independent claims.

2. Dependent Claims 2, 4 and 6-21:

Appellants respectfully submit that the dependent claims are patentable for the same reasons as the independent claims, as well as for their own additional features. The patentability of various of these claims is described below.

a. Dependent Claims 7, 12 and 17:

As one example, dependent claims 7, 12 and 17 specifically indicate that the individual prospective member state comprises a sequence number of the prospective member, and the comparing compares the sequence number of the prospective member with a sequence number of the processing group. Appellants respectfully submit that there is no description, teaching or suggestion in Moiin of a sequence number. As understood, a sequence is defined, for instance, as a continuity of progression (see, e.g., Webster's Ninth New Collegiate Dictionary). Appellants use the sequence number to identify, for instance, the particular version of the group configuration. Appellants respectfully submit that Moiin is silent as to a sequence number. That is, there is no description in Moiin of a sequence number or of use of a sequence number in the join protocol.

Support for the rejection of claims 7, 12 and 17 is indicated at Col. 5, lines 24-26 and 32-46 and Col. 6, lines 13-17 of Moiin. However, a careful reading of those sections fails to describe a sequence number. Instead, various fields are described without reference to sequence. For instance, an identification field is described which uniquely describes the node. This

identification field, however, is not described as having a particular sequence or of being a sequence number. Further, this field is neither compared nor updated. Additionally, a cluster vector field and a cluster vector size are described. These fields describe which members are in the cluster and the size of the cluster, but have no reference to sequence. These fields do not describe a sequence number of the prospective member or of the group, as claimed by appellants. Moiin fails to describe a sequence number or a sequence number used to join a member to a group. Based on the foregoing, appellants respectfully submit that dependent claims 7, 12 and 17 are not anticipated by Moiin, and respectfully request reversal of the rejection of those claims based on Moiin.

b. Dependent Claims 8, 13 and 18:

As another example, dependent claims 8, 13, and 18 specifically recite that the sequence number of the prospective member is less than the sequence number of the processing group, and wherein the updating comprises updating the sequence number of the prospective member with the sequence number of the processing group. Appellants respectfully submit that there is no description, teaching or suggestion in Moiin of a sequence number, nor is there any description, teaching or suggestion in Moiin of appellants' recited elements in which the sequence number is less than the sequence number of a processing group and that the updating includes updating the sequence number of the prospective member with the sequence number of the processing group. Moiin is silent as to these features.

Support for the rejection of claims 8, 13, and 18 is indicated at Col. 5, lines 36-38 of Moiin. However, a careful reading of that section merely describes a cluster size field that indicates the number of nodes included in the cluster. There is no description of a sequence number and there is no description of updating the sequence number of the prospective member with the sequence number of the processing group, when the sequence number of the prospective member is less than the sequence number of the processing group. This is simply missing from Moiin. Based on the foregoing, appellants respectfully submit that dependent claims 8, 13 and

18 are not anticipated by Moiin, and appellants respectfully request a reversal of the rejection of those claims.

c. Dependent Claims 9, 14 and 19:

As yet another example, dependent claims 9, 14 and 19 recite determining an activity status of the processing group prior to the updating of the sequence number, wherein the updating of the sequence number of the prospective member includes updating if the processing group is active. Again, appellants respectfully submit that there is no description, teaching or suggestion in Moiin of a sequence number; nor of determining an activity status of the processing group prior to updating the sequence number; nor of updating the sequence number of the prospective member if the processing group is active. Again, appellants respectfully submit that Moiin is silent as to these features.

Support for the rejection of claims 9, 14 and 19 is not provided in the Office Action. Instead, the Office Action merely states that Moiin teaches this. Appellants respectfully disagree. Based on the foregoing, appellants respectfully submit that dependent claims 9, 14 and 19 are not anticipated by Moiin, and appellants respectfully request a reversal of the rejection of those claims.

d. Dependent Claims 11, 16 and 21:

In a further example, dependent claims 11, 16 and 21 specifically recite wherein the updating of at least a portion of the state of the processing group after the joining includes updating the sequence number of the processing group. Again, appellants respectfully submit that there is no description, teaching or suggestion in Moiin of a sequence number, nor is there any description, teaching or suggestion that the updating of the at least a portion of the state of the processing group after the joining includes updating the sequence number of the processing group. Moiin is silent as to this feature because Moiin does not use sequence numbers in its determination. Instead, Moiin broadcasts messages to determine those nodes that are to be

included in a cluster. There is absolutely no discussion of a sequence number or of using a sequence number in the processing of the join protocol of Moiin.

Support for this rejection is indicated at Col. 5, lines 36-38. However, these lines merely describe a cluster size field which indicates the size of the cluster. There is no description of any of the elements claimed by appellants. Based on the foregoing, appellants respectfully submit that dependent claims 11, 16 and 21 are not anticipated by Moiin, and appellants respectfully request a reversal of the rejection of those claims.

e. Dependent Claim 49:

Further, dependent claim 49 specifically states that the sequence number of the prospective member identifies a version of a proposed processing group to join. There is no description, teaching or suggestion in Moiin of a sequence number nor of a sequence number identifying a version of a processing group. Moiin does not need such a sequence number, since the join protocol of Moiin relies on receiving reconfiguration messages from potential cluster members. It does not rely on nor compare sequence numbers, as claimed by appellants.

Support for this rejection in indicated as being implicit at Col. 7, lines 41-61. However, that section of Moiin makes no reference to sequence numbers. That section merely indicates that the messages are compared. A comparison of messages is not an implicit or explicit teaching of a sequence number or that a sequence number identifies a version of a proposed group to join. Messages can include information other than sequence numbers that is compared, as in the case of Moiin. Thus, appellants respectfully disagree that a comparison of messages implicitly teaches a comparison of sequence numbers, in which a sequence number indicates a version of a processing group.

Based on the foregoing, appellants respectfully submit that claim 49 is not described, taught or suggested by Moiin, and appellants respectfully request an indication of allowability for dependent claim 49.

3. Independent Claims 28, 35 and 42:

In this aspect, appellants' invention is directed to joining a prospective member to an inactive processing group. As one particular example, appellants claim a method of managing processing groups of a distributed computing environment (e.g., independent claim 28). The method includes, for instance, joining a prospective member to an inactive processing group; comparing at least a portion of an individual prospective member state with at least a portion of a group state of the processing group, the individual prospective member state comprising state defined for the individual prospective member; and updating the at least a portion of the group state. Thus, in this aspect of appellants' claimed invention, a prospective member is joining an inactive processing group. Further, at least a portion of an individual prospective member state is compared with at least a portion of a group state, the individual prospective member state comprising state defined for the individual prospective member. This is very different from the teachings of Moiin.

The join protocol of Moiin does not make a distinction as to whether the cluster being joined by a node is an inactive cluster or an active cluster. There is no such test being performed in the join protocol, as depicted in FIG. 4 and described in Cols. 5-7 of Moiin. Moiin merely broadcasts reconfiguration messages to all nodes which are potentially members of a new cluster, regardless of each node's membership in any current clusters (see, e.g., Col. 5, lines 60-65). Moiin then waits for reply messages and compares the received replies. Appellants respectfully submit that there is no discussion in Moiin of whether the cluster is active or inactive. For at least this reason, appellants respectfully submit that Moiin does not anticipate appellants' claimed invention.

Additionally, as described above, appellants respectfully submit that there is no description, teaching or suggestion in Moiin of appellants' claimed element of comparing at least a portion of an individual prospective member state with at least a portion of a group state of the processing group, the individual prospective member state comprising state defined for the

individual prospective member. The comparisons described in Moiin are not comparisons between individual prospective member state and group state. Thus, appellants respectfully submit that Moiin does not anticipate independent claim 28, as well as independent claims 35 and 42, and appellants respectfully request reversal of the rejection of those claims based on Moiin.

4. Dependent Claims 29-34, 36-41, and 43-48:

Appellants respectfully submit that these dependent claims are patentable for the same reasons as the independent claims, as well as for their own additional features. The patentability of various of these claims is described below.

a. Dependent Claims 29, 36 and 43:

As one example, dependent claims 29, 36 and 43 specifically indicate that the individual prospective member state comprises a sequence number of the prospective member, and the comparing compares the sequence number of the prospective member with the sequence number of the group state. Appellants respectfully submit that there is no description, teaching or suggestion in Moiin of a sequence number, nor is there any description, teaching or suggestion in Moiin of appellants' recited element of comparing the sequence number of the prospective member with the sequence number of the group state. Moiin is silent as to these features.

Support for the rejection of claims 29, 36 and 43 is indicated at Col. 5, lines 24-26 and 32-46 and Col. 6, lines 13-17 of Moiin. However, a careful reading of those sections fails to describe a sequence number or the elements claimed by appellants. Instead, various fields are described without reference to sequence, as described above. Thus, appellants respectfully submit that claims 29, 36 and 43 are patentable over Moiin, and appellants respectfully request reversal of the rejection of those claims based on Moiin.

b. Dependent Claims 30, 37 and 44:

As another example, dependent claims 30, 37 and 44 specifically recite that the updating of the sequence number of the group state with the sequence number of the prospective member is performed if the sequence number of the prospective member is smaller than the sequence number of the group state. Appellants respectfully submit that these claims are patentable for the same reasons as dependent claims 8, 13 and 18 described above, as well as for their own additional features. Based on the foregoing, appellants respectfully submit that dependent claims 30, 37 and 44 are patentable over Moiin, and respectfully request reversal of this rejection based on Moiin.

c. Dependent Claims 31, 38 and 45:

As yet another example, dependent claims 31, 38 and 45 recite updating the sequence number of the group state with a highest sequence number of the members of the processing group, if a quorum of the processing group exists. Again, appellants respectfully submit that there is no description, teaching or suggestion in Moiin of a sequence number nor of updating the sequence number of the group state with the highest sequence number of the members of the processing group, if a quorum of the processing group exists. Moiin is silent as to these features.

Support for this rejection is indicated at Col. 6, lines 13-17 of Moiin. However, a careful reading of this section merely describes the updating of a next cluster size field and next cluster vector to represent a cluster of the nodes for which a reconfiguration message was received. There is no discussion in that section or any other section of Moiin of a sequence number or of updating the sequence number of the group state with the highest sequence number of the members of the processing group if a quorum of the processing group exists. Thus, appellants respectfully submit that dependent claims 31, 38 and 45 are not anticipated by Moiin, and appellants respectfully request a reversal of the rejection of those claims based on Moiin.

d. Dependent Claims 33, 40 and 47:

In a further example, dependent claims 33, 40 and 47 specifically recite wherein the activating comprises updating a local copy of the group state for any member of the processing group whose sequence number is less than a current sequence number of the processing group. Again, appellants respectfully submit that there is no description, teaching or suggestion in Moiin of a sequence number, nor is there any description, teaching or suggestion that the activating includes updating a local copy of the group state for any member of the processing group whose sequence number is less than a current sequence number of the processing group.

Support for this rejection is indicated at Col. 1, lines 60-61 and Col. 6, lines 13-17. However, Col. 1, lines 60-61 merely state that nodes of a cluster which has a quorum are never in disagreement regarding the state of the cluster. This is not a description of sequence numbers or of updating the group state for any member of the processing group whose sequence number is less than a current sequence number of the processing group. Further, Col. 6, lines 13-17 also fails to describe, teach or suggest this claimed element. Based on the foregoing, appellants respectfully submit that dependent claims 30, 40 and 47 are not anticipated by Moiin, and appellants respectfully request a reversal of the rejection of those claims based on Moiin.

e. Dependent Claims 34, 41 and 48:

In a further example, dependent claims 34, 41 and 48 specifically recite wherein the activating further includes changing the group state to active if a majority of the members of the processing group have a sequence number matching the current sequence number and none of the members has aborted. Again, this is not described, taught or suggested in Moiin. Moiin does not use sequence numbers, but instead, transmits reconfiguration messages to join a node to a cluster. There is no description, teaching or suggestion in Moiin of sequence numbers nor of changing a group state to active, nor of changing the group state to active if the majority of the

members of the processing group have a sequence number matching the current sequence number and none of the numbers has aborted.

The failure of Moiin to explicitly teach this is even admitted in the Office Action (see, e.g., p. 7, line 1 of the Final Office Action). However, it is indicated in the Office Action that such a claimed element is inherent, citing Col. 1, lines 60-61 of Moiin. However, these lines merely indicate that nodes of a cluster which has a quorum are not in disagreement regarding the state of the cluster. This is not a teaching of the specific aspects claimed by appellants. Appellants respectfully disagree that appellants' claimed element is inherent in the teaching that a quorum indicates that there is no disagreement. Many techniques may be used to determine if a quorum is met and these techniques do not necessarily include the use of sequence numbers.

The doctrine of inherency is well-founded in patent law and is best described in an excerpt from Hansgirg v. Kemmer, 26 C.C.P.A. 937, 102 F.2d 212, 40 U.S.P.Q. 665 (1939):

> Inherency, however, may not be established by probabilities or possibilities. The mere fact that a certain thing *may* result from a given set of circumstances is not sufficient. [citations omitted.] If, however, the disclosure [of the cited reference] is sufficient to show that the natural result flowing from the operation as taught would result in the performance of the questioned function, it seems to be well settled that the disclosure should be regarded as sufficient [to anticipate the claimed invention].

Id. at 940, 102 F.2d at 214, 40 U.S.P.Q. at 667; Stoller v. Ford Motor Co., 18 U.S.P.Q. 2d 1545, 1547 (Fed. Circ. 1991); Tyler Refrigeration v. Kysor Industrial Corporation, 227 U.S.P.Q. 845, 847 (Fed. Cir. 1985); Ex parte Levy, 17 U.S.P.Q. 2d 1461, 1464 (B.P.A. I. 1990); In re Oelrich and Divigard,. 212 U.S.P.Q. 323, 326 (C.C.P. A. 1981).

In Ex parte Levy, the court stated that "[i]n relying upon the theory of inherency, the examiner must provide a basis in fact and/or technical reasoning to reasonably support the determination that the allegedly inherent characteristic *necessarily* flows from the teachings of the applied prior art." Ex parte Levy, 17 U.S.P.Q. 2d at 1464 (lengthy citation omitted) (italics added).

Appellants respectfully submit that the Examiner has neither pointed to any passage in Moiin where the use of sequence numbers necessarily flows from the disclosure, nor set forth any technical reasons in support of an inherency rejection of the cited language. Thus, appellants respectfully submit that Moiin does not anticipate claims 34, 41 and 48, and appellants respectfully request reversal of this rejection.

II. Rejection under 35 U.S.C 102(e) Over US Patent No. 6,449,734

Claims 22-27:

Claims 22-27 stand rejected under 35 U.S.C. 102(e) as being anticipated by Shrivastava et al. (U.S. Patent No. 6,449,734). Appellants respectfully submit that the rejection of these claims is erroneous and respectfully request reversal of this rejection for the reasons below.

This aspect of appellants' invention is directed to handling failed members. As one example, appellants claim a method of managing processing groups of a distributed computing environment (e.g., in independent claim 22), in which the method includes detecting a failure of at least one member of a processing group; quiescing activity to a group state of the processing group; and updating at least a portion of the group state in order to exclude the at least one member of the processing group. The updating includes updating a sequence number of the group state, the sequence number identifying a version of the processing group. Thus, in this aspect of appellants' claimed invention, when a failure of a member of a processing group is detected, the processing group is updated in order to exclude that member. The updating

includes updating the sequence number that identifies a version of the processing group. This is very different from the teachings of Shrivastava.

Although Shrivastava uses similar words, such as sequence number or failure, Shrivastava still does not describe, teach or suggest one or more aspects of appellants' claimed invention. In appellants' claimed invention, a sequence number identifies a version of the processing group (e.g., cluster) and the sequence number is updated in order to exclude a member of the processing group. In contrast, in Shrivastava, a sequence number is used to identify a particular transaction (see, e.g., Col. 13, lines 37-45) and not to describe the version of a processing group. Thus, the sequence number of Shrivastava is very different from the sequence number of appellants' claimed invention.

Further, the sequence number in Shrivastava is not updated when a transaction is not added to the group of transactions. Instead, the same transaction number is used and it is the duplicate sequence number that indicates that a transaction failed (see, e.g., Abstract; Col. 123, lines 53-49). The failure or exclusion is not indicated by an update of the group state, as claimed by appellants. There is no update in Shrivastava. Instead, the same, non-updated number is used.

Moreover, although various references are made in Shrivastava of removing a system from a cluster (see, e.g., Col. 5, lines 53-57; Col. 9, lines 32-33), Shrivastava does not describe how to remove that system from the cluster. Instead, it focuses on how to keep the transaction consistent. There is absolutely no teaching or suggestion in Shrivastava of detecting a failure of a member of a processing group and updating group state in order to exclude that member from the processing group, wherein the updating comprises updating a sequence number of the group, in which the sequence number identifies a version of the group. Since this is missing from Shrivastava, Shrivastava fails to describe, teach or suggest appellants' claimed invention.

For at least the above reasons, appellants respectfully submit that independent claim 22, as well as independent claims 24 and 26, and any claims that depend therefrom are patentable over Shrivastava. Accordingly, appellants respectfully request reversal of the rejection of claims 22-27 based on Shrivastava.

Conclusion

Appellants respectfully request reversal of the §102(e) rejections of claims 1-49 set forth in the Final Office Action. Appellants respectfully submit that their claimed invention is not anticipated by Moiin or Shrivastava.

For all of the above reasons, appellants allege error in rejecting their claims as anticipated based on the applied art. Accordingly, reversal of all rejections is respectfully requested.

Blanche E. Schiller

Blanche E. Schiller
Attorney for Appellants
Registration No.: 35,670

Dated: November 1, 2004

HESLIN ROTHENBERG FARLEY & MESITI P.C.
5 Columbia Circle
Albany, New York 12203-5160
Telephone: (518) 452-5600
Facsimile: (518) 452-5579

Appendix A

1. A method of managing processing groups of a distributed computing environment, said method comprising:

comparing at least a portion of an individual prospective member state of a prospective member of a processing group with at least a portion of a group state of the processing group, said individual prospective member state comprising state defined for the individual prospective member;

updating said at least a portion of the individual prospective member state, should said comparing indicate a difference; and

joining said prospective member to said processing group, in response to said updating.

2. The method of claim 1, further comprising:

quiescing activity to said group state; and

retrieving said group state after quiescing activity, for use in said comparing.

3. A system of managing processing groups of a distributed computing environment, said system comprising:

means for comparing at least a portion of an individual prospective member state of a prospective member of a processing group with at least a portion of a group state of the processing group, said individual prospective member state comprising state defined for the individual prospective member;

means for updating said at least a portion of the individual prospective member state, should said comparing indicate a difference; and

means for joining said prospective member to said processing group, in response to said updating.

4. The system of claim 3, further comprising:

means for quiescing activity to said group state; and

means for retrieving said group state after quiescing activity, for use in said comparing.

5. At least one program storage device readable by a machine, tangibly embodying at least one program of instructions executable by the machine to perform a method of managing processing groups of a distributed computing environment, said method comprising:

comparing at least a portion of an individual prospective member state of a prospective member of a processing group with at least a portion of a group state of the processing group, said individual prospective member state comprising state defined for the individual prospective member;

updating said at least a portion of the individual prospective member state, should said comparing indicate a difference; and

joining said prospective member to said processing group, in response to said updating.

6. The at least one program storage device of claim 5, further comprising:

quiescing activity to said group state; and

retrieving said group state after quiescing activity, for use in said comparing.

7. The method of claim 1, wherein the individual prospective member state comprises a sequence number of the prospective member, and wherein the comparing comprises comparing the sequence number of the prospective member with a sequence number of the processing group.

8. The method of claim 7, wherein the sequence number of the prospective member is less than the sequence number of the processing group, and wherein the updating comprises updating the sequence number of the prospective member with the sequence number of the processing group.

9. The method of claim 8, further comprising determining an activity status of the processing group prior to the updating, wherein updating the sequence number of the prospective member comprises updating if the processing group is active.

10. The method of claim 1, further comprising updating at least a portion of the state of the processing group after the joining.

11. The method of claim 10, wherein the updating at least a portion of the state of the processing group after the joining comprises updating the sequence number of the processing group.

12. The system of claim 3, wherein the individual prospective member state comprises a sequence number of the prospective member, and wherein the means for comparing comprises means for comparing the sequence number of the prospective member with a sequence number of the processing group.

13. The system of claim 12, wherein the sequence number of the prospective member is less than the sequence number of the processing group, and wherein the means for updating comprises means for updating the sequence number of the prospective member with the sequence number of the processing group.

14. The system of claim 13, further comprising means for determining an activity status of the processing group prior to the updating, wherein the means for updating the sequence number of the prospective member comprises means for updating the sequence number of the prospective member if the processing group is active.

15. The system of claim 3, further comprising means for updating at least a portion of the state of the processing group after the joining.

16. The system of claim 15, wherein the means for updating at least a portion of the state of the processing group after the joining comprises means for updating the sequence number of the processing group.

17. The at least one program storage device of claim 5, wherein the individual prospective member state comprises a sequence number of the prospective member, and wherein the comparing comprises comparing the sequence number of the prospective member with a sequence number of the processing group.

18. The at least one program storage device of claim 17, wherein the sequence number of the prospective member is less than the sequence number of the processing group, and wherein the updating comprises updating the sequence number of the prospective member with the sequence number of the processing group.

19. The at least one program storage device of claim 18, further comprising determining an activity status of the processing group prior to the updating, wherein updating the sequence number of the prospective member comprises updating if the processing group is active.

20. The at least one program storage device of claim 5, further comprising updating at least a portion of the state of the processing group after the joining.

21. The at least one program storage device of claim 20, wherein the updating at least a portion of the state of the processing group after the joining comprises updating the sequence number of the processing group.

22. A method of managing processing groups of a distributed computing environment, the method comprising:

detecting a failure of at least one member of a processing group;

quiescing activity to a group state of the processing group; and

updating at least a portion of the group state in order to exclude the at least one member of the processing group, wherein the updating comprises updating a sequence number of the group state, said sequence number identifying a version of the processing group.

23. The method of claim 22, wherein the quiescing and updating are performed if the processing group is active and the at least one member of the processing group comprises less than a majority of the processing group.

24. A system of managing processing groups of a distributed computing environment, the system comprising:

means for detecting a failure of at least one member of a processing group;

means for quiescing activity to a group state of the processing group; and

means for updating at least a portion of the group state in order to exclude the at least one member of the processing group, wherein the means for updating comprises means for updating a sequence number of the group state, said sequence number identifying a version of the processing group.

25. The system of claim 24, wherein the means for quiescing and means for updating are performed if the processing group is active and the at least one member of the processing group comprises less than a majority of the processing group.

26. At least one program storage device readable by a machine tangibly embodying at least one program of instructions executable by the machine to perform a method of managing processing groups of a distributed computing environment, the method comprising:

detecting a failure of at least one member of a processing group;

quiescing activity to a group state of the processing group; and

updating at least a portion of the group state in order to exclude the at least one member of the processing group, wherein the updating comprises updating a sequence number of the group state, said sequence number identifying a version of the processing group.

27. The at least one program storage device of claim 26, wherein the quiescing and updating are performed if the processing group is active and the at least one member of the processing group comprises less than a majority of the processing group.

28. A method of managing processing groups of a distributed computing environment, the method comprising:

joining a prospective member to an inactive processing group;

comparing at least a portion of an individual prospective member state with at least a portion of a group state of the processing group, said individual prospective member state comprising state defined for the individual prospective member; and

updating the at least a portion of the group state.

29. The method of claim 28, wherein the individual prospective member state comprises a sequence number of the prospective member, and wherein the comparing comprises comparing the sequence number of the prospective member with a sequence number of the group state.

30. The method of claim 29, wherein the updating comprises updating the sequence number of the group state with the sequence number of the prospective member if the sequence number of the prospective member is smaller than the sequence number of the group state.

31. The method of claim 29, wherein the updating comprises updating the sequence number of the group state with a highest sequence number of the members of the processing group if a quorum of the processing group exists.

32. The method of claim 28, further comprising activating the processing group.

33. The method of claim 32, wherein the activating comprises updating a local copy of the group state for any member of the processing group whose sequence number is less than a current sequence number of the processing group.

34. The method of claim 33, wherein the activating further comprises changing the group state to active if a majority of the members of the processing group have a sequence number matching the current sequence number and none of the members has aborted.

35. A system of managing processing groups of a distributed computing environment, the system comprising:

means for joining a prospective member to an inactive processing group;

means for comparing at least a portion of an individual prospective member state with at least a portion of a group state of the processing group, said individual

prospective member state comprising state defined for the individual prospective member; and

means for updating the at least a portion of the group state.

36. The system of claim 35, wherein the individual prospective member state comprises a sequence number of the prospective member, and wherein the means for comparing comprises means for comparing the sequence number of the prospective member with a sequence number of the group state.

37. The system of claim 36, wherein the means for updating comprises means for updating the sequence number of the group state with the sequence number of the prospective member if the sequence number of the prospective member is smaller than the sequence number of the group state.

38. The system of claim 36, wherein the means for updating comprises means for updating the sequence number of the group state with a highest sequence number of the members of the processing group if a quorum of the processing group exists.

39. The system of claim 35, further comprising means for activating the processing group.

40. The system of claim 39, wherein the means for activating comprises means for updating a local copy of the group state for any member of the processing group whose sequence number is less than a current sequence number of the processing group.

41. The system of claim 40, wherein the means for activating further comprises means for changing the group state to active if a majority of the members of the processing group have a sequence number matching the current sequence number and none of the members has aborted.

42. At least one program storage device readable by a machine tangibly embodying at least one program of instructions executable by the machine to perform a method of managing processing groups of a distributed computing environment, the method comprising:

joining a prospective member to an inactive processing group;

comparing at least a portion of an individual prospective member state with at least a portion of a group state of the processing group, said individual prospective member state comprising state defined for the individual prospective member; and

updating the at least a portion of the group state.

43. The at least one program storage device of claim 42, wherein the individual prospective member state comprises a sequence number of the prospective member, and wherein the comparing comprises comparing the sequence number of the prospective member with a sequence number of the group state.

44. The at least one program storage device of claim 43, wherein the updating comprises updating the sequence number of the group state with the sequence number of the prospective member if the sequence number of the prospective member is smaller than the sequence number of the group state.

45. The at least one program storage device of claim 43, wherein the updating comprises updating the sequence number of the group state with a highest sequence number of the members of the processing group if a quorum of the processing group exists.

46. The at least one program storage device of claim 42, further comprising activating the processing group.

47. The at least one program storage device of claim 46, wherein the activating comprises updating a local copy of the group state for any member of the processing group whose sequence number is less than a current sequence number of the processing group.

48. The at least one program storage device of claim 47, wherein the activating further comprises changing the group state to active if a majority of the members of the processing group have a sequence number matching the current sequence number and none of the members has aborted.

49. The method of claim 7, wherein the sequence number of the prospective member identifies a version of a proposed processing group to join.